Solving SAT by Symport/Antiport P Systems with Membrane Division

Artiom Alhazov

Institute of Mathematics and Computer Science
Academy of Science of Moldova
Str. Academiei 5, Chişinău, MD 2028, Moldova

Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain
E-mail: artiom@math.md, artiome.alhazov@estudiants.urv.es

Summary. We present an $O(n) + O(\log m)$-time solution of SAT with n variables and m clauses by a uniform family of deterministic P systems with communication rules (antiport-2/1 and antiport-1/2) and membrane division rules (without polarization) and unstructured environment. Nothing is sent to the environment except yes or no, in one copy. We can even start with the empty environment if we also use symport-1 rules.

1 Introduction

Membrane systems are a (biologically inspired) theoretical framework of distributed parallel multiset processing. See [6] for the bibliography of the domain.

This paper is a continuation of [5], where membrane division rules were added to (tissue) P systems with symport and antiport, to solve SAT in a uniform way. Here we improve the results from [5] in the following:

- Tissue P system is replaced by a ("usual") P system with tree-like structure.
- The P system gives the result in a time which depends on the number of clauses logarithmically, not linearly.
- The computation is now deterministic, not just confluent.
- The cardinality of the environment alphabet E is only 1.
- Only antiport-2/1 and antiport-1/2 are used, while the construction from [5] used symport-1, and antiports 1/2, 2/1, 3/2.
- Nothing is sent into environment except the result, just like P systems with active membranes ([4]).
- Paying the price of additionally using symport-1 we no longer need to bring any objects from the environment.

The determinism can be reached due to the massive parallelism (what could happen in either order should happen simultaneously) and the system does not need resources (supply of objects) from the environment because the number of objects can grow via membrane division.

2 Symport/Antiport P Systems with Membrane Division

A symport/antiport P system is defined as a tuple $\Pi = (O, E, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m, R, i_0)$, where O is a finite set of objects, $E \subseteq O$ is a set of objects present in the environment in the unbounded quantities, μ is a membrane structure with m regions. w_i, $1 \le i \le m$, are the strings representing initial multisets of each region i. R_i, $1 \le i \le m$, are the rules associated to each membrane i. The communicative rules are of one of the following forms: (u, in), (v, out), $(u, out; v, in)$, $u, v \in O^+$ (the first two forms are called symport, while the latter is called antiport). The membrane division rules are of the form $[_h\ a\ ]_h \to [_h\ b\ ]_h [_h\ c\ ]_h$.

The rules are applied non-deterministically, in a maximally parallel manner. The system is deterministic if there is only one computation possible (for any reachable configurations, all configurations reachable in one step are indistinguishable). Notice that this property does not imply that there is only one rule applicable for every object because different copies of the same object are indistinguishable, and so are different membranes with the same label and contents.

3 Solving SAT

The problem is defined as follows: given a boolean formula

$$\begin{aligned} \gamma &= C1 \wedge \ldots \wedge C_m, \text{ where} \\ C_i &= y_{i,1} \vee \ldots \vee y_{i,k_i},\ 1 \le i \le m, \\ &\quad y_{i,j} \in \{x_k, \neg x_k \mid 1 \le k \le n\},\ 1 \le i \le m,\ 1 \le j \le k_i \end{aligned}$$

find whether γ has a solution.

We define M as $Ceil(\log_2 m)$ $(= \min\{j \in \mathbb{N} \mid 2^j \ge m\})$; this is a number, logarithmic with respect to the number of clauses, which will be needed for defining the system below (notice that $2^M < 2m$). Look at the set S defined below (this is a set of objects that we will need in the environment in a sufficient number of copies to perform the clause evaluation in all membranes in parallel). Notice that $|S| = 6n+2n+(M+1)n+2^M+M+1 = n(M+9)+2^M+M+1 = O(nM)+O(m)$. We also define N as $Ceil(\log_2 |S|)$ $(= \min\{j \in \mathbb{N} \mid 2^j \ge |S|\})$; this is a number logarithmic with respect to $|S|$ (notice that $2^N < 2|S| = O(nM) + O(m)$).

We define T as $6n + 2N + 2M + 3$ (the time we claim is enough to produce object *yes* in the skin membrane if and only if γ is satisfiable). The instance of a problem is encoded in the alphabet $\Sigma = \{s_{i,j}, s'_{i,j} \mid 1 \le i \le n, 1 \le j \le m\}$ by objects $s_{i,j}$ if clause j contains x_i, and $s'_{i,j}$ if clause j contains $\neg x_i$. We construct the following P system:

$$\begin{aligned}
\Pi &= (O, E = \{\$\}, \mu, w_0, w_1, w_2, w_3, w_4, R_0, R_1, R_2, R_3, R_4, R),\\
O &= \{\$, f, d, a_1, \texttt{yes}, \texttt{no}\} \cup \{e_i \mid 0 \le i \le 2n+N\}\\
&\quad \cup \{d_{i,0} \mid 0 \le i < n+N\} \cup \{d_{2n+N+i,j} \mid 0 \le i \le N, 0 \le j \le 2^i\}\\
&\quad \cup \{t'_i, f'_i \mid 1 \le i \le n\} \cup \{b_i \mid 0 \le i \le T\} \cup \Sigma,\\
S &= \{t''_i, t'''_i, t''''_i, f''_i, f'''_i, f''''_i, \mid 1 \le i \le n\}\\
&\quad \cup \{a_i \mid 2 \le i \le 2n+1\} \cup \{t_{i,j}, f_{i,j} \mid 1 \le i \le n, 0 \le j \le M\}\\
&\quad \cup \{c_{2^j(2i),2^j(2i+1)} \mid 0 \le j \le M, 0 \le i < 2^{M-j-1}\}\\
&\quad \cup \{z_j \mid 0 \le j < M\} \cup \{z\} \text{ are synonyms}\\
&\qquad \text{of some symbols } d_{2n+2N,j}, 0 \le j \le 2^N,\\
\mu &= [_0\, [_1\ ]_1 [_2\ ]_2 [_3\ ]_3 [_4\ ]_4 [_5\ ]_5\,]_0,\\
w_0 &= e_{n+N}f,\ w_1 = a_1 d,\ w_2 = \texttt{yes},\ w_3 = b_0\texttt{no},\ w_4 = e_0 e_{n+N},\ w_5 = d_{0,0}z.
\end{aligned}$$

The following rules are used:

- **Cloning objects e_{n+N}**

E1 $[_4\ e_i\ ]_4 \to [_4\ e_{i+1}\ ]_4 [_4\ e_{i+1}\ ]_4 \in R$, $0 \le i < n+N$;
E2 $(e_{n+N}e_{n+N}, out; e_{n+N}, in) \in R_4$.

Starting from $[_4\ e_0 e_{n+N}\ ]_4$, in $n+N$ steps, 2^{n+N} copies of $[_4\ e_{n+N}e_{n+N}\ ]_4$ are produced by rule E1. Then, starting from one copy of e_{n+N}, in $n+N$ more steps, 2^{n+N} copies of e_{n+N} are produced by rule E2.

- **Producing objects for the computation**

O1 $[_5\ d_{i,0}\ ]_5 \to [_5\ d_{i+1,0}\ ]_5 [_5\ z\ ]_5 \in R$, $0 \le i < n+N$;
O2 $[_5\ d_{i,0}\ ]_5 \to [_5\ d_{i+1,0}\ ]_5 [_5\ d_{i+1,0}\ ]_5 \in R$, $n+N \le i < 2n+N$;
O3 $[_5\ d_{i,j}\ ]_5 \to [_5\ d_{i+1,2j}\ ]_5 [_5\ d_{i+1,2j+1}\ ]_5 \in R$,
$2n+N \le i < 2n+2N$, $0 \le j \le 2^N$;
O4 $(d_{2n+2N,j}z, out; e_{n+N}, in) \in R_1$, $0 \le j < 2^N$.

In membrane 5, the object $d_{i,0}$ "waits" for $n+N$ steps, i.e., changes into $d_{n+N,0}$ by rules O1 (also $n+N$ dummy membranes are produced). In another n steps, from $[_5\ d_{n+N,0}z\ ]_5$, rules O2 produces 2^n copies of $[_5\ d_{2n+N,0}z\ ]_5$. In further N steps, each of these 2^n membranes divides by rule O3 and produces $[_5\ d_{2n+2N,0}z\ ]_5, \ldots, [_5\ d_{2n+2N,2^N-1}z\ ]_5$, i.e., 2^N membranes with different objects. These are the objects which we will need in the skin membrane for the further computation, they are simultaneously brought in the skin by rule O4 (let us give, for simplicity, pseudonyms to objects $d_{2n+2N,j}$ for different j from entire set S: $d_{2n+2N,0} = t'_1, d_{2n+2N,1} = t'_2, \ldots, d_{2n+2N,|S|} = z$).

- **Variable assignments**

A1 $[_1\ a_i\ ]_1 \to [_1\ t'_i\ ]_1 [_1\ f'_i\ ]_1 \in R$, $1 \le i \le n$;

A2 $(t'_i, out; t''_i a_{i+1}, in) \in R_1$,
$(f'_i, out; f''_i a_{i+1}, in) \in R_1$, $1 \le i \le n$;
A3 $(a_{n+i} t''_i, out; t'''_i, in) \in R_1$,
$(a_{n+i} f''_i, out; f'''_i, in) \in R_1$, $1 \le i \le n$;
A4 $(t'''_i, out; t_{i,0} a_{n+i+1}, in) \in R_1$,
$(f'''_i, out; f_{i,0} a_{n+i+1}, in) \in R_1$, $1 \le i \le n$;
A5 $(t_{i,j}, out; t_{i,j+1} t_{i,j+1}, in) \in R_1$,
$(f_{i,j}, out; f_{i,j+1} f_{i,j+1}, in) \in R_1$, $1 \le i \le n$, $0 \le j < M$.

Using rules A1 and A2, $[_1\ a_i\ ]_1$ changes to $[_1\ t''_{i+1} a_{i+1}\ ]_1 [_1\ f''_{i+1} a_{i+1}\ ]_1$ in two steps, except for the case $i = 1$, where objects t'_1 and f'_1 wait for $2n + 2N$ steps until the objects from S are produced and moved to the environment. In $2n$ steps, rules A3 and A4 exchange objects t''_i in membranes labelled 1 for objects $t_{i,0}$ (and f''_i for $f_{i,0}$). After some object $t_{i,0}$ ($f_{i,0}$) appears in membrane 1, it is exchanged for 2^M copies of $t_{i,M}$ ($f_{i,0}$, respectively), this replication takes another $2n$ steps. In this way, there will be enough copies of $t_{i,M}$ (and $f_{i,0}$), i.e., witnesses of true/false assignment of variable x_i for each object $s_{i,j}$ (and $s'_{i,j}$), encoding the fact that if x_i is true (false, respectively), then clause C_j is satisfied. This part of computation will finish in at most $6n + 2N + M$ steps from the beginning of the computation.

- **Checking clauses**

C1 $(t_{i,M} s_{i,j}, out; c_{j-1,j}, in) \in R_1$, $1 \le i \le n$;
C2 $(c_{2^j(2i),2^j(2i+1)} c_{2^j(2i+1),2^j(2i+2)}, out; c_{2^j(2i),2^j(2i+2)}, in) \in R_1$,
$0 \le j \le M$, $0 \le i \le \frac{m}{2^j \cdot 2} - 1$,
$(c_{2^j(2i),2^j(2i+1)}, out; z_j c_{2^j(2i),2^j(2i+2)}, in) \in R_1$, $0 \le j < M$, $\frac{m}{2^j \cdot 2} - 1 < i \le \frac{2^M}{2^j \cdot 2} - 1$;
C3 $(c_{0,2^M} d, out; z, in) \in R_1$;
C4 $(\texttt{yes}, out; fd, in) \in R_2$;
C5 $[_3\ b_i\ ]_3 \to [_3\ z\ ]_3 [_3\ b_{i+1}\ ]_3 \in R$, $0 \le i < T$;
C6 $(b_T \texttt{no}, out; f, in) \in R_3$;
C7 $(\texttt{yes}, out; \$\$, in) \in R_0$, $(\texttt{no}, out; \$\$, in) \in R_0$.

All clauses are checked simultaneously by rules C1: clause C_j satisfied corresponds to object $c_{j-1,j}$. Then rules C2 "assemble" the clause satisfiability: objects c_{j_1,j_2} and c_{j_2,j_3} are replaced by an object c_{j_1,j_3}, but this assembly has a binary character: $j_1 = 2^j(2i)$, $j_2 = 2^j(2i+1)$, $j_3 = 2^j(2i+2)$ ($c_{0,1} + c_{1,2} \longrightarrow c_{0,2}$, $c_{2,3} + c_{3,4} \longrightarrow c_{2,4}$, $c_{0,2} + c_{2,4} \longrightarrow c_{0,4}$, etc.) The second group of rules provide "automatic satisfiability" of the non-existing clauses $m + 1, \ldots, 2^M$.

In this way in every membrane corresponding to a solution of γ, an object $c_{0,2^M}$ will be obtained. Then it will come in the skin membrane together with an object d using rule C3. Next, d will be exchanged by yes by C4, also removing f from the skin (if and only if γ is satisfiable, and at most one copy).

This will happen at $M + 3$ steps after the end of the variable assignment phase (this number is exact because all clauses need to be checked and contribute to this result), i.e., at step $6n + 2N + 2M + 3 = T$. Meanwhile, in membrane 3 a counting until T is performed by rules C5. Rule C6 is executed if and only if the object f is still in the skin (i.e., when γ is not satisfiable).

Then, either yes or no will be brought into the environment by C7. The system halts in time $T+1$ if the formula is satisfiable, and in time $T+2$ otherwise.

Replacing rules C7 by $(\texttt{yes}, out)$) and $(\texttt{no}, out)$ and redefining E as $\emptyset$ would lead to an equivalent solution.

4 Final Remarks and Open Questions

The satisfiability problem for a boolean formula in the disjunctive normal form with n variables and m clauses can be solved in time $O(n) + O(\log m)$ by a uniform family of deterministic P systems with communication rules (antiport-2/1, antiport-1/2, symport-1) and membrane division rules (without polarization) and empty environment. The use of symport rules can be eliminated at a price of starting with (at least 2 copies of) one symbol in the environment, and the system will only eject the result in the environment.

We would like to make the following comments:

- The determinism heavily depends on the massive parallelism (not just in different membranes corresponding to different clauses, but also in each membrane for the same clause). Is massive parallelism of rules with respect to membranes (i.e., using for some membrane a number of rules not bounded by a constant independent of n and m) really needed in order to have a deterministic construction, or it is only needed for a construction that runs in logarithmic time with respect to the number of clauses?
- Increasing both the number of membranes and objects heavily depends on using membrane division. Since in the symport/antiport model the objects can be taken from the environment, it would be interesting to consider a variant of membrane systems where membrane division does not increase the number of objects. For instance, can SAT be solved by symport/antiport P systems with membrane separation? (See [1] for the definition of membrane separation.)
- We expect that the number of starting membranes can be decreased. Is a solution as above possible starting with two membranes?

Acknowledgements

The author is supported by the project TIC2002-04220-C03-02 of the Research Group on Mathematical Linguistics, Tarragona. The author also acknowledges the Moldovan Research and Development Association (MRDA) and the U.S. Civilian Research and Development Foundation (CRDF), Award No. MM2-3034.

A special thanks is addressed to Gheorghe Păun for encouraging the author to write this article.

References

1. A. Alhazov, T.-O. Ishdorj: Membrane operations in P systems with active membranes. In *Proceedings of the Second Brainstorming Week on Membrane Computing* (Gh. Păun, A. Riscos, A. Romero, F. Sancho, eds.), Report RGNC 01/04, University of Seville, 2004, 37-44.
2. A. Păun, Gh. Păun: The power of communication: P systems with symport/antiport. *New Generation Computing*, 20, 3 (2002), 295–306.
3. Gh. Păun: *Computing with Membranes: An Introduction*. Springer-Verlag, Berlin, 2002.
4. Gh. Păun: P systems with active membranes: Attacking NP-complete problems. *J. Automata, Languages and Combinatorics*, 6, 1 (2001), 75–90, and CDMTCS TR 102, Univ. of Auckland, 1999.
5. Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez: Tissue P systems with cell division. In *Proceedings of the Second Brainstorming Week on Membrane Computing* (Gh. Păun, A. Riscos, A. Romero, F. Sancho, eds.), Report RGNC 01/04, University of Seville, 2004, 380–386.
6. The P systems web page: `http://psystems.disco.unimib.it`.